

Classical Model Predictive Control of a Permanent Magnet Synchronous Motor

Jean-François Stumper, Alexander Dötlinger and Ralph Kennel

KEYWORDS

Optimal Control, Control of Drive, Synchronous Motor.

ADDRESS

Institute of Electrical Drive Systems and Power Electronics,
Technische Universität München, Germany.

ABSTRACT

A model predictive control (MPC) scheme for a permanent-magnet synchronous motor (PMSM) is presented. The torque controller optimizes a quadratic cost consisting of control error and machine losses repeatedly, accounting the voltage and current limitations. The scheme extensively relies on optimization, to meet the runtime limitation, a suboptimal algorithm based on differential flatness, continuous parameterization and linear programming is introduced.

The multivariable controller exploits cross-coupling effects in the long-range constrained predictive control strategy. The optimization results in fast and smooth torque dynamics while inherently using field-weakening to improve the power efficiency and the current dynamics in high speed operation. As distinctive MPC feature, constraint handling is improved, instead of just saturating the control input, field weakening is applied dynamically to bypass the voltage limitation. The performance of the scheme is demonstrated by experimental and numerical results.

I. INTRODUCTION

The efforts of implementing predictive controllers in electrical drives aim at replacing the classical cascaded field-oriented control structure with PI controllers. The machine can be better exploited by improved control behavior, the system variables are optimized. In this contribution, the conventional torque and current control structure of two separate controllers is changed to multi-input multi-output (MIMO) control. By transformation into the field-oriented frame, torque generation is decoupled from flux variation, however, the current dynamics are still strongly coupled, therefore a MIMO controller is advantageous. The aimed improvements are better current and voltage constraint handling by exploiting cross-coupling between the orthogonal components, and better power efficiency and dynamics by optimally adjusting the currents in both dynamic and steady-state operation.

The major obstacle in implementing predictive control schemes is the limited computational power, inherited by the high sampling rates. The most widespread schemes trade computational feasibility against compromises in the problem formulation, for instance, generalized predictive control

(GPC) has a high prediction horizon but is unconstrained, whereas predictive torque control (PTC) is constrained but so far only reaches 2 steps of prediction [1]. To obtain the advantages claimed by classical MPC on constrained MIMO systems, both, inclusion of constraints and a high prediction horizon are required. Only few schemes so far satisfy these two requirements [2]. Using continuous control in the field-oriented frame, the analytical problem description enables using efficient optimization algorithms to maximize the obtained information for a given computational power.

The online solution of the linearly constrained linear-quadratic problem, typical for MPC, requires quadratic programming (QP) algorithms, which are, however, computationally too expensive for drive systems. A recent development is the use of explicit MPC, where an offline solution is computed and stored as look-up table in the real-time controller [3], [4]. The scheme reaches 5 prediction steps with constraints. Recently an online algorithm based on a fast-gradient algorithm was proposed for reference tracking control of a grid-connected inverter [5]. While the runtime results are convincing, experiments are pending. The advantage of online optimization in MPC is the possibility to manipulate or adapt parameters, which results in simplified commissioning, an aspect that is becoming more and more important in the recent developments [6].

This implementation entitled 'classical MPC' relies extensively on online-optimization. It is embedded in a flatness-based predictive control scheme [7], [8], where trajectory optimization and prediction are decoupled, and the trajectory generation can be simplified [9]. A suboptimal trajectory generation algorithm [10] is applied which is based on a continuous approach, the variables are not discretized but represented as a polynomial with undetermined coefficients. The cost function is linearized in the unconstrained optimum in order to use a linear programming (LP) solver, which is amongst the simplest and fastest numerical optimizers. As result, a (suboptimal) prediction of 2 ms with current and voltage constraints is obtained at 8 kHz sampling rate. This way, by solving the computational problem without compromising the MPC formulation, the merits of this controller can be studied.

The initial results have been presented in [11].

II. PROBLEM STATEMENT

A. Machine Model

In a first step the machine model is linearized. This will result in a linear-quadratic optimization problem with linear constraints, a very common problem in optimal control systems [12] that is also simpler to solve in real-time. Assuming

that the rotor speed does not change too much over the optimization horizon T ,

$$\frac{d}{dt}\omega_M(t) \approx 0 \Rightarrow \omega_M(t) = \text{const.} \quad \forall t \in [0, T], \quad (1)$$

the PMSM model and the voltage equations become linear. The assumption is justified if the current control loop is faster than the speed control loop. The electrical subsystem of the machine, consisting of the quadrature and direct currents i_q and i_d (peak values), is given as

$$L_d \frac{d}{dt} i_d = -R i_d + n_p \omega_M L_q i_q + u_d, \quad (2)$$

$$L_q \frac{d}{dt} i_q = -R i_q - n_p \omega_M L_d i_d - n_p \omega_M K + u_q, \quad (3)$$

$$\tau_M = \frac{3}{2} n_p K i_q. \quad (4)$$

The nomenclature is shown in table II in the appendix. It is noted that the reluctance torque $\tau_M^R = \frac{3}{2} n_p (L_d - L_q) i_d i_q$ is neglected, as this term is very small compared to the electromagnetic torque in surface-mounted PMSMs or in machines with small saliency. Furthermore, it would render the model nonlinear, requiring nonlinear optimization methods [13].

B. Optimization Goals and Cost Functional

The formulation of a suitable cost functional is a key point in predictive control, as it is the only tuning possibility of the control scheme. The optimization is aiming at minimizing the control error for good dynamical performance as well as machine losses for better efficiency. Both goals are included in the cost functional. By choosing the cost functional and weights well, it is possible to find a good trade-off between both goals during transients, and eventually to fulfill both goals in steady-state. The cost functional for the predictive torque controller is

$$J = \int_0^T (P_{ctrl}(t) + W_L \cdot P_{loss}(t)) dt + T \cdot P_{ctrl}(T) \quad (5)$$

which trades off the squared control error from the constant torque reference τ_M^*

$$P_{ctrl}(t) = (\tau_M - \tau_M^*)^2 \quad (6)$$

with machine losses

$$\begin{aligned} P_{loss}(t) &= \frac{3}{2} R (i_d^2 + i_q^2) + \frac{3}{2} n_p \omega_M k_{Fe} (\Psi_d^2 + \Psi_q^2) \\ &= \frac{3}{2} R (i_d^2 + i_q^2) + \frac{3}{2} n_p \omega_M k_{Fe} ((L_d i_d + K)^2 + (L_q i_q)^2). \end{aligned} \quad (7)$$

The first term in P_{loss} represents copper losses, and the second term represents the iron losses consisting of hysteresis losses. The last term in (5) is the end-weight of the control error, it serves to improve convergence within finite time, especially when constraints are active. Its effect is shown in the results section. Eddy current losses are negligible on the tested machines, however, they could be included using the model presented in [14]. The machine losses can be reduced by field-weakening, where a trade-off between copper and iron

losses is found [15]. Imposing a negative direct current i_d , the flux magnitude in the stator is reduced while the copper losses increase. As the iron loss constant k_{Fe} is not part of standard motor parameters, it has to be determined experimentally [16]. The iron losses are quite considerable, for instance, the applied 2.64 kW machine has about 43 W copper losses and 200 W iron losses in rated operation, therefore the parameter k_{Fe} can be determined using only standard equipment. Inverter losses remain undiscussed as the switching frequency is arbitrary fixed.

There are two tuning parameters, W_L and T . The weight W_L was set 0.05, the value was determined heuristically. The optimization horizon is set $T = 2$ ms such that the cost functional includes the complete setpoint change. It is important that the optimization horizon is high enough, otherwise the open-loop and closed-loop trajectories differ and the behavior is strongly suboptimal. This is illustrated in Fig. 1 (left). The open-loop trajectory, a continuous trajectory optimal with respect to J , is generated and a first part is applied to the system. At the next sampling step (indicated by a circle), a new open-loop trajectory is generated and again its first part is applied to the system. The resulting closed-loop trajectory, which is the sequence of the discrete points at each sampling interval, differs significantly from the open-loop trajectories, especially if the horizon is too small. Then, the closed-loop trajectories simply do not fit the cost functional anymore and are suboptimal. For a horizon higher than required for the setpoint change, as shown in Fig. 1 (right), the difference between open- and closed-loop trajectories becomes smaller. Ideally the points should be superposed with the continuous open-loop trajectories, then, the closed-loop trajectories can be assumed optimal regarding the open-loop cost functional J .

C. Current and Voltage Constraints

The most important nonlinearities of a PMSM, in terms of control, are the voltage and current limitations. The current constraints prevent overheating of the machine, and the voltage is limited by the maximum output voltage of the voltage source inverter. The voltage constraints limit rotor speed as well as current dynamics in high-speed operation. Both constraints are approximated by affine equations, in order to be computationally efficiently treated.

The current range for the direct current i_d is limited to $i_d^{min} \leq i_d \leq 0$. Only negative values of i_d are desirable, as they improve power efficiency and reduce the induced voltage by weakening the flux magnitude in the stator [4], [15]. The lowest value i_d^{min} is the optimum value at rated speed ($\frac{\partial}{\partial i_d} P_{loss} = 0$) and is given as

$$i_d^{min} = -\frac{L_d K}{L_d^2 + R(n_p \omega_M n k_{Fe})^{-1}}, \quad (8)$$

which is independent of quadrature current i_q if the reluctance torque and the saliency are neglected, what is acceptable for surface-mounted PMSMs [14]. The value is doubled to enable further field-weakening to improve dynamics in high speed, an effect described in the results section. For the quadrature

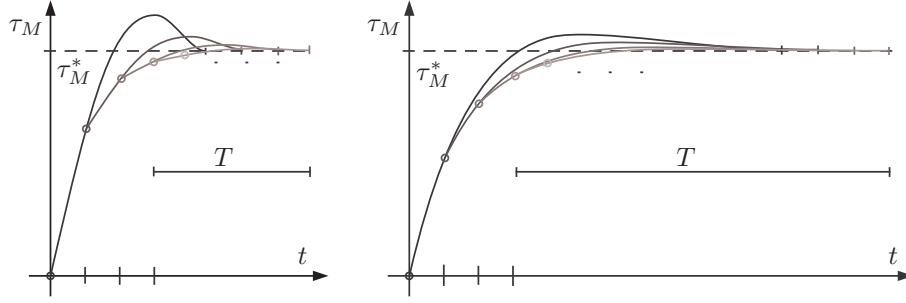


Fig. 1. Exemplary torque setpoint change to describe open- and closed-loop trajectories in long-range predictive control. Left: small horizon, right: high horizon. Circles: starting points for each trajectory planning iteration.

current i_q , the largest possible range of values should be available. The resulting linear constraints, shown in Fig. 2, almost completely fill the current region of interest. A representation with affine inequality constraints is thus acceptable.

The approximation of the voltage constraints is a bit more difficult. The q -axis should not be restricted, as the induced voltage is aligned to it and is the largest value that will appear. A steady-state analysis of the system equations (2), (3) shows that a rectangular voltage area results

$$Ri_d^{min} - n_p L_q \omega_M^{max} i_q^{max} \leq u_d \leq n_p L_q \omega_M^{max} i_q^{max}, \quad (9)$$

$$-Ri_q^{max} + n_p L_d \omega_M^{max} i_d^{min} - n_p K \omega_M^{max} \leq u_q \leq Ri_q^{max} + n_p K \omega_M^{max}. \quad (10)$$

This rectangle (dark grey on Fig. 2) is expanded such that the outer circle of the voltage limitation is hit (light grey on Fig. 2). During dynamical transients, the voltage vector points to one of the outer corners, subsequently touching the outer limiting circle. Therefore, a linear approximation of the voltage limits as a rectangle by the presented method, as shown on Fig. 2, does not limit the steady-state operational range and only marginally affects dynamics. A less restrictive method is presented in [5], where a time-varying constraint in form of a hexagon in stator frame is proposed. The limiting circle can be expanded to a hexagon by using overmodulation techniques [17]. While this is possible too with the underlying predictive control algorithm, the method in the (d, q) -frame is chosen for simplicity and to prevent possible current ripples.

III. OPTIMAL CONTROL ALGORITHM

To study real-time applicability of the presented scheme, first, the highest possible amount of optimization parameters is determined. The fastest optimizer with constraints is the widely known linear programming (LP) method. Table I shows some worst-case computational results of LP (simplex method from [18]) as function of the number of free parameters (CPU: 1.4 GHz industrial PC). More parameters lead to a higher number of iterations which are also more complex; the worst-case number of iterations is the number of parameters plus the number of constraints [12]. As in the underlying application,

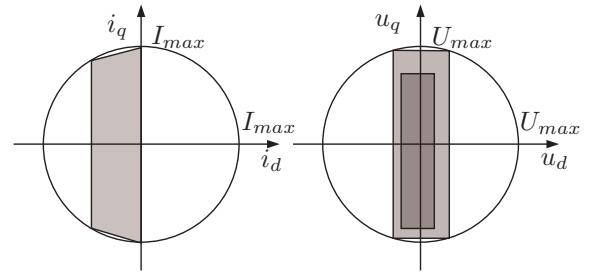


Fig. 2. Affine approximation of the current and voltage constraints. Circle: feasible set of current and voltage vectors, grey: feasible set after approximation of the constraints.

the constraints are decoupled, however, this worst-case is not to be expected. The maximum runtime is given by the sampling rate minus latency of input/output, therefore at 8 kHz sampling rate, it must be less than about 110 μ s. Thus, at best, 12 parameters can be optimized if a LP method is used. Runtime of the predictive controller is further discussed at the end of the section.

TABLE I
RUNTIME OF A LINEAR PROGRAM FOR SOME WORST-CASE PROBLEMS ON A 1.4 GHz CPU

Parameters	Constraints	Iterations	Runtime [μ s]
20	44	67	769
12	28	34	165
8	20	10	35

A. Trajectory Generation

The trajectory generation algorithm presented in [10], a development related to flatness-based methods [9], is chosen. It can optimize a quadratical cost function with linear constraints. As major differences to standard algorithms, it is applying a continuous parameterization instead discretization, and the computationally efficient linear programming solver is used instead of quadratic programming or iterative gradient search.

Even though LP is used, it is still quadratic optimization; the unconstrained solution to the quadratic cost is calculated first, then, constraints are included with the LP solver.

The trajectories for the current are defined as degree n power series with undetermined coefficients α_{ij} ,

$$i_d(t) = \sum_{k=0}^n \alpha_{dk} \frac{t^k}{T^k}, \quad i_q(t) = \sum_{k=0}^n \alpha_{qk} \frac{t^k}{T^k}, \quad t \in [0, T]. \quad (11)$$

This definition reduces the dimensionality of the generated trajectories rather than their length and is referred to as Ritz parameterization [12]. It is an alternative parameterization to the typical Euler discretization. The first coefficients α_{d0} and α_{q0} are the initial conditions, and the remaining 6 coefficients are determined by optimization. A high prediction horizon is obtained for a relatively small number of parameters. Due to the analyzed computational limitations, $n = 3$ is chosen as polynomial degree. Fig. 3 illustrates the computational advantage, with 3 parameters at 8 kHz sampling rate, using a discrete description, the prediction horizon is 0.375 ms, but with a degree 3 polynomial, a well-conditioned setpoint change can be described over the desired prediction horizon of 2 ms. A more complex trajectory may not be expected for the underlying application.

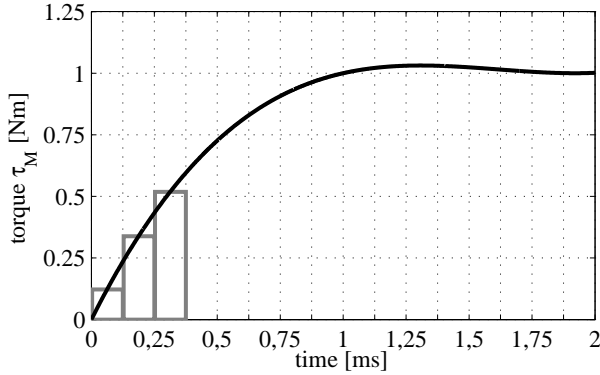


Fig. 3. A trajectory described by 3 free parameters. Black: polynomial-based curve, grey: discrete-time curve. Discrete-time horizon is 0.375 ms, continuous polynomial-based trajectory is well-conditioned at the desired horizon length 2 ms.

The corresponding voltages $u_{dq}(t)$ are computed by algebraic differentiation of (11) and by solving for the model equations (2) and (3), this is also called the flatness-based approach [19]. This way, the voltages do not need to be represented by additional parameters, and equality constraints are avoided in the optimization. The variables i_{dq} , $\frac{d}{dt}i_{dq}$ and u_{dq} are substituted in the cost functional J (5) by the found functionals.

The cost functional J is then a quadratic function of the unknown parameters α and of the parameters which were assumed constant, namely the motor parameters, the measured currents, the speed ω_M and the torque reference τ_M^* . Defining the vector of undetermined coefficients as $\alpha =$

$(\alpha_{d1}, \alpha_{d2}, \alpha_{d3}, \alpha_{q1}, \alpha_{q2}, \alpha_{q3})^T$, it is written as

$$J = \alpha^T Q \alpha + q^T \alpha + q_0. \quad (12)$$

Because of the parameterization with a polynomial basis, convexity must be discussed [9], in (5) the weight matrix was $\mathbb{R}^{2 \times 2}$ whereas now it is extended to $Q \in \mathbb{R}^{6 \times 6}$. Convexity guarantees the existence of a unique global minimum, and convergence of the optimization algorithm in finite time. The proof that convexity is maintained with this transformation is given in [10]. Graphically, for a simple example with two parameters, J can be represented as in Fig. 4 (left).

The inequality constraints are also parameterized with the polynomial. Exact parameterization requires linear matrix inequality (LMI) methods, which are however too involved for this application. A simpler way is to sample the trajectories for $i_{dq}(t)$ and $u_{dq}(t)$ at an interval $\frac{T}{n}$. In [10] it is proven that if an additional interlay is added, the transformation guarantees maintenance of the original constraints. For instance, a constraint $i_d(t) \leq 0 \forall t \in [0, T]$ is parameterized as

$$i_d(0) \leq 0, \quad (13)$$

$$i_d(k\frac{T}{n}) - \Delta i_d(0) \leq 0 \quad k = 1..3, \quad (14)$$

where $\Delta = 0.064$ is the respective interlay constant for $n = 3$ (see [10]). The first constraint on the initial condition does not need to be included, and the remaining 3 conditions are affine functions of α , but not of t , such that they can directly be included in linear-quadratic optimization in the parameter space α .

As J is convex, the unconstrained global optimum α_0^* is found algebraically by solving first-order necessary conditions,

$$\alpha_0^* = -\frac{1}{2}Q^{-1}q. \quad (15)$$

Then, by an affine coordinate transformation

$$\beta = A(\alpha - \alpha_0^*), \quad (16)$$

the problem can be reformulated as least-distance problem, i.e. a quadratical cost describing the distance to the unconstrained optimum. The linear transformation includes a shift of the origin of the coordinates, a coordinate rotation as well as a coordinate scaling, and is found with

$$A^T A = Q, \quad (17)$$

which can be solved with the Cholesky decomposition $A^T = \text{cholesky}(Q^T)$. As result, the cost functional looks much simpler and reduces to a sum of squares

$$J = \beta^T \beta, \quad (18)$$

and can be represented as in Fig. 4 (middle). The unconstrained optimum is thereby $\beta_0^* = 0$. This constrained least-distance problem is already simpler to solve than the original problem. In the next step, the least-distance problem is linearized around the unconstrained optimum, see Fig. 4 (right).

The squares in the cost function are replaced by absolute values

$$J = \beta^T \beta = \sum_i \beta_i^2 \approx \sum_i |\beta_i| = J'. \quad (19)$$

In the LP standard form, furthermore, only positive parameters are possible, therefore the variables are replaced by $\beta_i = \beta_{ip} - \beta_{in}$, with $\beta_{ip}, \beta_{in} \geq 0$. The absolute value can then be replaced by $|\beta_i| = \beta_{ip} + \beta_{in}$. Equivalence is guaranteed by minimizing the (positive) sum, such that at least one variable of each pair (β_{ip}, β_{in}) will be zero [12]. The linearization of the cost function inherits a large error in the value of J , but the values of the coefficients β are not affected that much: the least-distance problem is not so much different in the linear form as it would have been in the quadratical form. Furthermore, see that a difference only appears if a constraint is active, the unconstrained optimum is the same. Suboptimality is evaluated in the results section based on numerical simulations. The linear constraints on α are as well transformed directly with (16). Therefore, after all the transformations, the problem is available in standard form for linear programming, and a simplex solver [18] can be run. The optimal solution β^* has to be retransformed to find the optimum in the original coordinates

$$\alpha^* = \alpha_0 + A^{-1}\beta^*, \quad (20)$$

which is used in (11) and the motor model to obtain the optimal trajectories for the currents and voltages.

B. Implementation Aspects

All computations of the previous section are done using a computer algebra tool (Maplesoft Maple). The matrices for the LP solver are generated using the C code generation toolbox. The real-time software thus consists of a simplex tableau assignment, consisting of the initialization of one matrix and two vectors of floating-point variables, which is automatically generated code, a simplex LP solver from [18], and some post-processing. As the assignment is based on symbolic calculations, the motor parameters can be changed online. However, the majority of parameters is defined as constant, the compiler can then optimize code size as well as runtime of the initialization stage.

Alternatives to this procedure were evaluated. The direct use of a quadratic program is too slow, for instance, with 6 parameters, 16 (voltage) constraints and 20 iterations it requires 730 μs - if additionally current constraints were included, it would take even longer. If the problem is first transformed to least-distance, the number of iterations and the runtime reduce to about 50%.

Also online calculation of the trajectory generation algorithm instead of computer algebra generated code was analyzed. The matrix inversion with 6 parameters takes 2.5 μs and the cholesky decomposition 40 μs . Additionally this requires the use of a scientific mathematics library.

The choice to use a computer algebra tool and automatic code generation turns out as the easiest way to implement and

as well as computationally most efficient. The automatically generated code is copy-and-pasted to the real-time software, only a small LP solver like [18] has to be added. The chosen LP implementation, however, seems to be a somewhat slow implementation, runtime improvements are possible.

C. Predictive Control

The control structure is shown in Fig. 5. A cascaded control structure is chosen as speed is assumed constant for trajectory generation. As the mechanical plant is generally only roughly known, this overlying speed controller is advantageous. Model-based control can be used for the electrical subsystem of the motor as the parameters are known, but for the mechanical part, any robust feedback controller can be chosen.

The trajectory generation scheme is embedded in a predictive controller and repeated at every sampling step to generate the optimal control input $u_{dq}[k]$.

Additionally, for steady-state accuracy, a disturbance observer is necessary, for instance to compensate modeling errors of the induced voltage terms. Even though the induced voltage is modeled, variations in K caused for instance by heating or identification errors cause an offset of the current i_q . The approach is to assume the disturbance as constant and calculate it directly based on the MPC model and past control inputs and current measurements [20]. The respective delays have to be accounted in the design.

D. Timing

First, measurement and control timing is analyzed. Two timing sequences are shown in Fig. 6 (a) and (b). The interrupt-based control system triggers an interrupt every 125 μs (signal 3). At this instant, the applied voltage command (signal 2) is modulated in the power electronics. At the same instant, the A/D conversion of the current measurements (signal 1) is performed to avoid the impact of current ripples. It is seen that to calculate the command $u_q[k]$, only the current $i_q[k-1]$ is available. Furthermore, once the command $u_q[k]$ is calculated, it needs to wait until the next interrupt. This delay of one step is denoted *computational delay* and is accounted with a delay compensation technique. A prediction of one sampling step with the model, the current $i_q[k-1]$ as well as the previously commanded voltage $u_q[k-1]$ is applied to generate the delay-compensated current $i_q[k|k-1]$ as initial condition for the control law to compute $u_q[k]$, see [21]. Furthermore, see that the response to the commanded voltage $u_q[k]$ is $i_q[k+1]$ which is available one interrupt later, this is the *plant delay* and it is naturally included in the predictive controller by recalculating the trajectory at every sampling step. From the predicted trajectory, the applied voltage is $u_{dq}[k] = u_{dq}(0)$.

From the timing sequences, interesting insight into the computational demands of the algorithm is gained. The first part of the controller signal in Fig. 6 shows the calculation time for the simplex tableau initialization, it takes about 10 μs . Included in these calculations, which is automatically generated code resulting from symbolical calculations, is a calculation of the

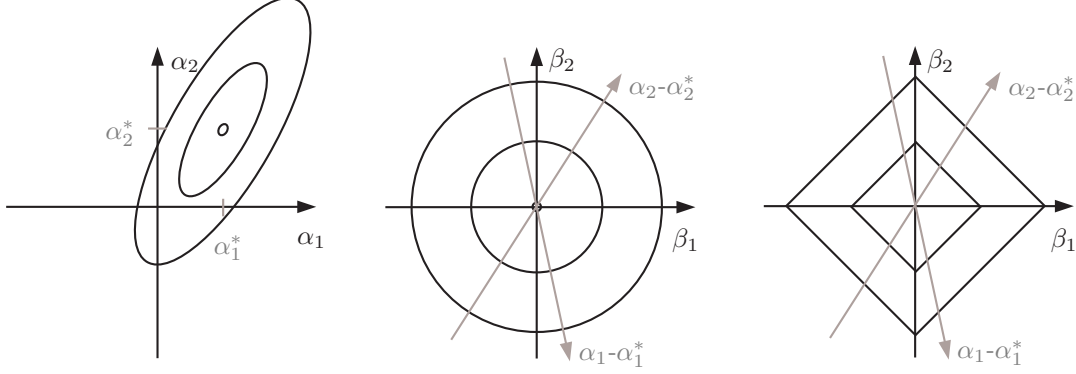


Fig. 4. Transformation and linearization of the optimization problem. Lines are equimagnitude contours of the cost function. Left: original problem, middle: transformed least-distance problem, right: transformed and linearized problem.

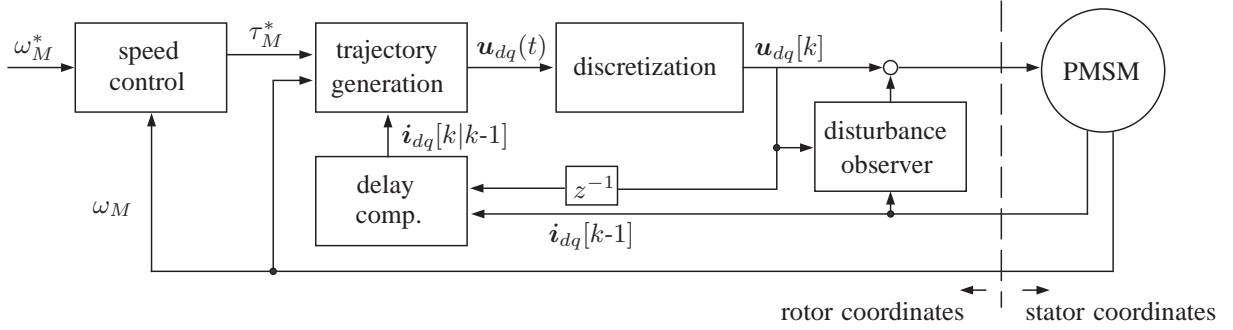


Fig. 5. Control structure of the predictive torque controller cascaded by PI speed control.

unconstrained optimum and the linearization of the problem. The second and biggest part of the controller signal is the runtime of the linear program [18]. At the beginning of Fig. 6 (a), where voltage and current are both zero, it is only about $20\mu\text{s}$, but to calculate the voltage step at 2000 rpm shown in Fig. 6 (b), more iterations are involved as many constraints are active, and the computation time rises to almost $60\mu\text{s}$. The total time of the interrupt handling, latency, the simplex initialization, the LP solver and the post-processing sum up to almost $100\mu\text{s}$ in the worst case, therefore up to 80% of the available time is used.

IV. RESULTS

A surface-mounted PMSM with parameters shown in table II is used. It is coupled to a load drive such that an arbitrary speed can be imposed. The algorithm is implemented on a PC-104 based real-time system with a 1.4 GHz CPU described in [22]. The voltage limitation was set to 250 V because of hardware limitations, accordingly, rated speed is reduced from 3000 to 2200 rpm.

Experimental results of the proposed scheme are shown in Fig. 7. Subfigure (a) shows the response to two subsequent

speed reference steps, the load drive is deactivated. The torque is increased rapidly by a high but feasible voltage peak - the PI speed controller can be made very fast such that also a step appears on the reference torque for the MPC controller. The direct current depends on the speed and thereby reduces iron losses which are considerable at high speeds. Losses are decreased by about 4%, and the efficiency is improved by about 0.5% at 2000 rpm. Better results are obtained on motors with higher inductances [15].

The next three subfigures (b), (c) and (d) show fast torque transients at zero, medium and high speed, respectively. The PMSM is in torque control mode while the load drive keeps speed constant at 0, 2000 or 2400 rpm, respectively. The current components are well decoupled, a fast current change on the quadrature axis does not affect the direct axis in (b) and (c). With two separate PI controllers, a short current excursion would be seen on the direct axis during the torque transient. Again, the current on the direct axis i_d is dependent on the speed. Furthermore, the torque change is fast and at the same time smooth, the voltage becomes smoothly small for smaller control errors – a nice characteristic of quadratic cost functionals, compared to linear cost functions which result

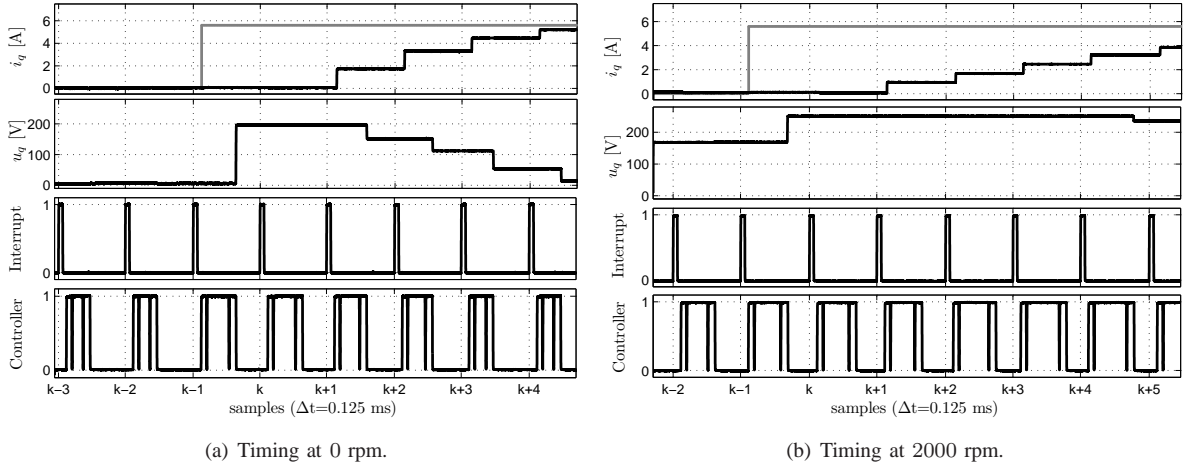


Fig. 6. Experimental timing results of the predictive control scheme. 1: quadrature current i_q without delay compensation, 2: quadrature voltage u_q , 3: interrupt handling, 4: control law computation (first part: initialization, second part: LP solver, third part: post-processing).

in deadbeat behavior and are more sensitive to uncertainties [21].

On subfigures (b) and (c), the behavior with active voltage constraint is the same as when using standard saturation or anti-windup strategies. On subfigure (d), however, a different behavior is seen, the direct current i_d is reduced to perform field-weakening. This implies that the stator induced voltage is reduced on the quadrature axis, see eq. (3). This mechanism is known from minimum-time current control [23]. With field-weakening, the gap between induced and maximum voltage increases, the derivative of the quadrature current $\frac{d}{dt}i_q$ is higher and the torque-generation dynamics are increased, at the cost of higher copper losses on the direct axis. Without additional field-weakening, the reference torque would not be reached after the optimization horizon of 2 ms, thereby the end-weight of the control error in J oversized the loss term. Therefore, in this predictive control implementation, field-weakening not only improves efficiency, but also improves dynamics by exploiting cross-coupling between the orthogonal current components to optimally bypass the voltage saturation.

The situation described above is the one where the highest number of constraints is active, and thus, also where the suboptimality of the proposed trajectory generation method is the highest. Based on the simulation results in Fig. 8, the suboptimality can be analyzed by comparing the results to a QP solver, which is not practicable in the experimental setup. In Fig. 8 (a) no real difference can be seen as only few constraints are active. In Fig. 8 (b) more constraints are active, as result, it is seen that the dynamical field-weakening is much more intense, but foremost, of longer duration when using QP. The simulations in Fig. 8 have a slightly different quantitative behavior as the experiments, caused by parameter uncertainties.

It is also possible to operate the PMSM beyond rated speed with steady field-weakening to bypass the voltage saturation on the quadrature axis, as shown in [4]. It is remarked that the current on the direct axis i_d has no reference, its value

is obtained from the optimization of the cost functional. Therefore the method works well and is numerically stable; the optimal value follows inherently.

V. CONCLUSION

Two questions have been central to this work, first, is it possible to implement linear model predictive control (MPC) with online optimization on an electrical drive, and second, will this advanced controller lead to any merits.

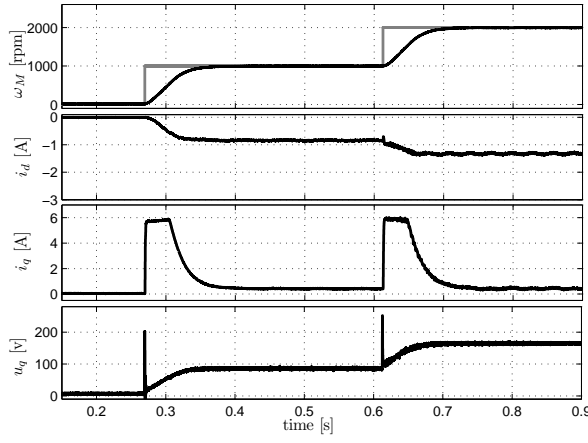
A model predictive control scheme for a PMSM was introduced. Based on suboptimal real-time optimization, the currents and voltages are computed according to a cost functional. The prediction horizon is 2 ms at a sampling rate of 8 kHz, and voltage and current constraints are respected. The work can be seen as first implementation of classical model predictive control on an electrical drive, where the aspects of constraint inclusion and high prediction horizon are satisfied while online optimization is used. Implementing online MPC is therefore possible even on fast-sampling systems such as electrical drives.

Additionally to the advantages of general predictive controllers, which are precise accounting for timing of measurement and control as well as respecting current and voltage constraints, the advantages of the long-range constrained predictive MIMO control scheme can be concluded as follows: improved accounting of cross-coupling, fast and smooth dynamical behavior, improved power efficiency by field weakening, and improved dynamics close to voltage saturation by field-weakening.

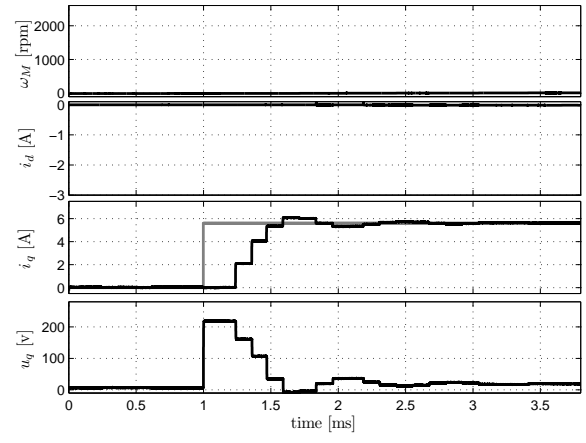
As negative point, the high demands to computational power must be named.

ACKNOWLEDGMENTS

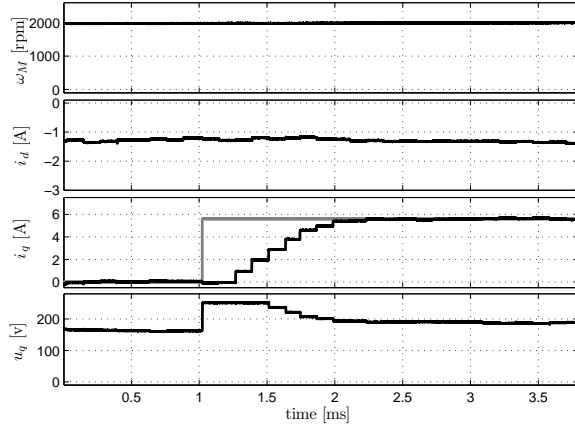
This work was supported through the National Research Fund of Luxembourg under grant PhD-08-070. The authors thank Janos Jung for his valuable help at the implementation stage and Sascha Khl for the fruitful discussions.



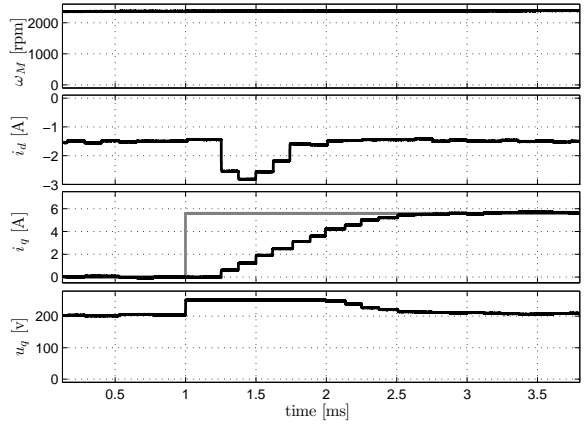
(a) Speed steps from 0 to 1000 to 2000 rpm.



(b) Torque step at zero speed.

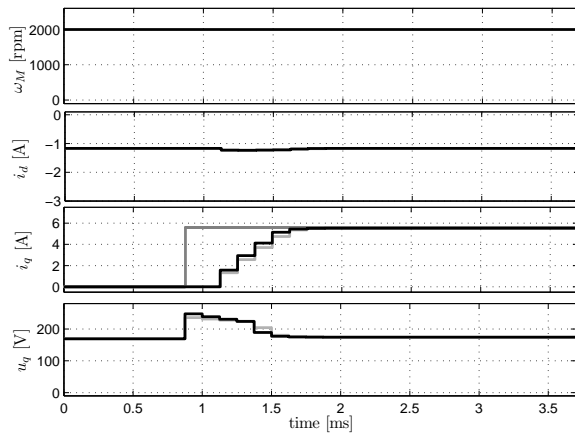


(c) Torque step at 2000 rpm.

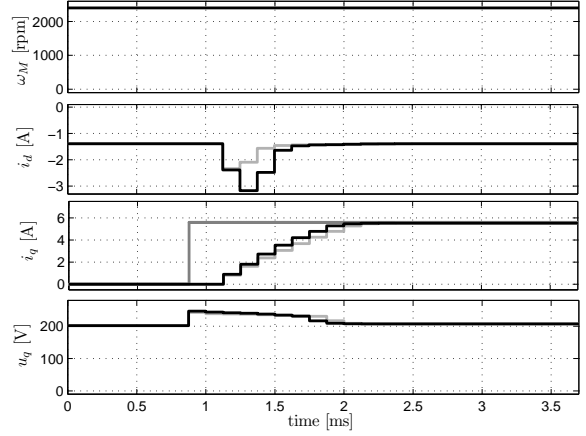


(d) Torque step at 2400 rpm.

Fig. 7. Experimental results of the predictive control scheme. 1: rotor speed ω_M , 2: direct current i_d without compensation, 3: quadrature current i_q without compensation, 4: quadrature voltage u_q .



(a) Torque step at 2000 rpm.



(b) Torque step at 2400 rpm.

Fig. 8. Simulation results of the predictive control scheme, black: QP, light grey: LP. 1: rotor speed ω_M , 2: direct current i_d without compensation, 3: quadrature current i_q without compensation, 4: quadrature voltage u_q .

REFERENCES

- [1] P. Cortés, M. Kazmierkowski, R. Kennel, D. Quevedo and J. Rodriguez, Predictive control in power electronics and drives, *IEEE Trans. Ind. Electron.*, Vol. 55, No. 2, pp. 4312–4324, 2008.
- [2] T. Geyer, G.A. Beccuti, G. Papafotiou and M. Morari, Model predictive direct torque control of permanent magnet syn-

chronous motors, *Energy Convers. Congress and Exp.*, pp. 199–206, 2010.

[3] S. Bolognani, R. Kennel, S. Kuehl and G. Paccagnella, Speed and current model predictive control of an IPM synchronous motor drive, *Proc. IEEE Int. Electric Machines and Drives Conf.*, pp. 1597–1602, 2011.

[4] S. Bolognani, S. Bolognani, L. Peretti and M. Zigliotto, Combined speed and current model predictive control with inherent field-weakening features for PMSM drives, *Proc. of the IEEE Med. Electrotechnical Conf.*, pp. 472–478, 2008.

[5] S. Richter, S. Mariéthoz and M. Morari, High-speed online MPC based on a fast gradient method applied to power converter control, *Proc. of the American Control Conf.*, pp. 4737–4743, 2010.

[6] J.H. Lee, Model predictive control: Review of three decades of development, *Int. J. Control, Automation and Systems*, Vol. 9, No. 3, pp. 415–424, 2011.

[7] M. Fliess and R. Marquez, Continuous-time linear predictive control and flatness: a module-theoretic setting with examples, *Int. J. of Control*, Vol. 73, No. 7, pp. 606–623, 2000.

[8] V. Hagenmeyer and E. Delaleau, Continuous-time non-linear flatness-based predictive control: an exact feedforward linearization setting with an induction drive example, *Int. J. of Control*, Vol. 81, No. 10, pp. 1645–1663, 2008.

[9] M. Guay and J.F. Forbes, Real-time dynamic optimization of controllable linear systems, *J. of Guidance, Control and Dynamics*, Vol. 29, No. 4, pp. 929–935, 2006.

[10] J-F. Stumper and R. Kennel, Computationally efficient trajectory optimization for linear control systems with input and state constraints, *Proc. of the American Control Conf.*, pp. 1904–1909, 2011.

[11] J-F. Stumper, A. Döttinger, J. Jung and R. Kennel, Predictive control of a permanent magnet synchronous machine based on real-time dynamic optimization, *Proc. Europ. Conf. on Power Electron. And Appl.*, paper 99, 2011.

[12] D.A. Pierre, *Optimization theory with applications*. New York: Wiley, 1969.

[13] E. Delaleau and A.M. Stanković, Flatness-based hierarchical control of the PM synchronous motor, *Proc. of the American Control Conf.*, pp. 65–70, 2004.

[14] C. Mademlis, J. Xypteras and N. Margaris, Loss minimization in surface permanent-magnet synchronous motor drives, *IEEE Trans. Ind. Electron.*, Vol. 47, No. 1, pp. 115–122, 2000.

[15] C. Cavallaro, A.O. Di Tommaso, R. Micelli, A. Raciti, G.R. Galluzzo and M. Trapanese, Efficiency enhancement of permanent-magnet synchronous motor drives by online loss minimization approaches, *IEEE Trans. Ind. Electron.*, Vol. 52, No. 4, pp. 1153–1160, 2005.

[16] N. Urasaki, T. Senjyu and K. Uezato, A novel calculation method for iron loss resistance suitable in modeling permanent-magnet synchronous motors, *IEEE Trans. Energy Convers.*, Vol. 18, No. 1, pp. 41–47, 2003.

[17] N.P. Quang and J.A. Dittich, *Vector control of three-phase AC machines: System development in the practice*. Berlin: Springer Verlag, 2008.

[18] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical recipes in C: the art of scientific computing, 2nd ed., Ch. 10.8. Cambridge: Cambridge University Press, 1992.

[19] H. Sira-Ramírez and S.K. Agrawal, *Differentially flat systems*, Ch. 8.2. New York: Marcel Dekker, 2004.

[20] K.H. Kim and M.J. Youn, A simple and robust digital current control technique of a PM synchronous motor using time delay control approach, *IEEE Trans. Power Electron.*, Vol. 16, No. 1, pp. 72–82, 2001.

[21] H-T. Moon, H-S. Kim and M-J. Youn, A discrete-time predictive current control for PMSM, *IEEE Trans. Power Electron.*, Vol. 18, No. 1, pp. 464–472, 2003.

[22] N.A. Ameen, A.A. Naassani, R. Kennel, Design of a digital system dedicated for electrical drive applications, *EPE J.*, Vol. 20, No. 4, pp. 37–44, 2010.

[23] J.W. Choi and S.K. Sul, Generalized solution of minimum time current control in three-phase balanced systems, *IEEE Trans. Power Electron.*, Vol. 45, No. 5, pp. 738–744, 1998.

APPENDIX: MOTOR PARAMETERS

TABLE II
NOMINAL PARAMETERS OF THE SYNCHRONOUS MOTOR

Manufacturer & Model	Merkes MT5 1050
Rated Power P_N	2760 W
Rated Torque τ_{MN}	10.5 Nm
Rated Current (peak)	8 A
Rated Speed ω_{MN}	3000 rpm
Pole Pairs n_p	3
Rated Voltage U_N (peak)	560 V resp. 330 V
Stator Inductance L_d, L_q	4.8, 7.2 mH
Stator Resistance R	0.92 Ω
Motor Constant K (peak)	0.334 Vs
Iron Loss Constant (Hysteresis) k_{Fe}	1.27 $\frac{A}{Vs}$